# Iterative Rounding algorithms for the generalised Gasoline Problem

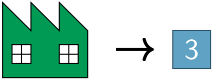
Lucas Lorieau

Grenoble INP, Ensimag - UGA

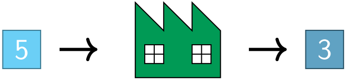
June 26, 2024

# Introductory example

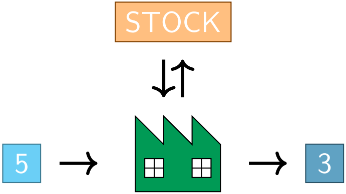
# Introductory example



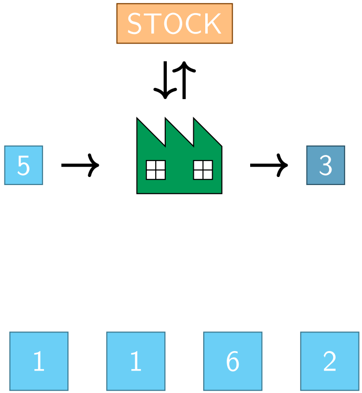
# Introductory example



# Introductory example



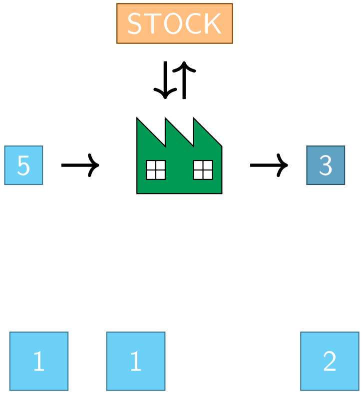
# Introductory example




| Day | + | - |
|---|---|---|
| 1 | | 5 |
| 2 | | 0 |
| 3 | | 2 |
| 4 | | 3 |

Objective : Minimize the span of the stock

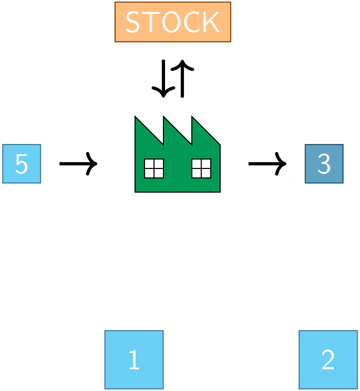
# Introductory example




| Day | + | - |
|---|---|---|
| 1 | 6 | 5 |
| 2 | | 0 |
| 3 | | 2 |
| 4 | | 3 |

Objective : Minimize the span of the stock

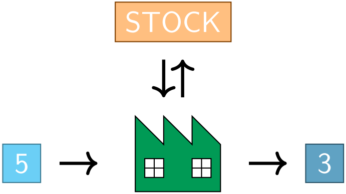
# Introductory example




| Day | + | - |
|---|---|---|
| 1 | 6 | 5 |
| 2 | 1 | 0 |
| 3 | | 2 |
| 4 | | 3 |

Objective : Minimize the span of the stock

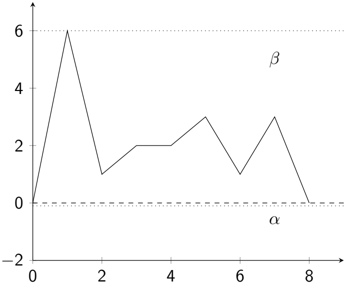
# Introductory example




| Day | + | - |
|---|---|---|
| 1 | 6 | 5 |
| 2 | 1 | 0 |
| 3 | 1 | 2 |
| 4 | 2 | 3 |

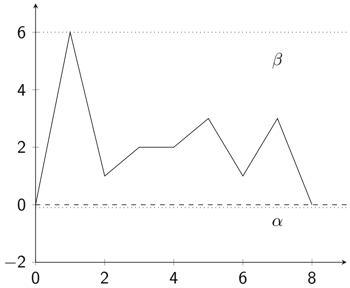
Objective : Minimize the span of the stock

# Representation of the stock over time




| Day | + | - |
|---|---|---|
| 1 | 6 | 5 |
| 2 | 1 | 0 |
| 3 | 1 | 2 |
| 4 | 2 | 3 |

# Definition of the Gasoline problem




**Instance** Refill orders X and Production orders Y with $\sum_i x_i = \sum_i y_i$

**Solution** A permutation (matrix) $Z$ of X

$$\min \ \beta - \alpha \quad \text{s.t.}$$

$$\sum_{j=1}^{n} \sum_{i=1}^{k} x_i z_{ij} - \sum_{i=1}^{k-1} y_i \leq \beta \qquad \text{for } 1 \leq k \leq n$$

$$\sum_{j=1}^{n} \sum_{i=1}^{k} x_i z_{ij} - \sum_{i=1}^{k} y_i \geq \alpha \qquad \text{for } 1 \leq k \leq n$$

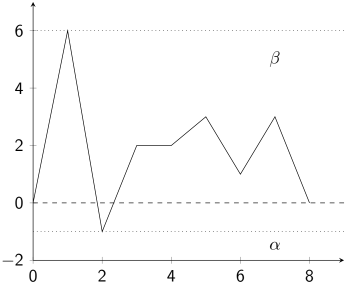
$$Z\mathbb{1} \leq \mathbb{1}$$

$$\mathbb{1}^T Z \leq \mathbb{1}^T$$

$$z_{ij} \in \{0, 1\} \qquad \text{for } 1 \leq i, j \leq n$$

$$\alpha, \beta \in \mathbb{R}$$

# Definition of the Gasoline problem




**Instance** Refill orders X and Production orders Y with $\sum_i x_i = \sum_i y_i$

**Solution** A permutation (matrix) $Z$ of X

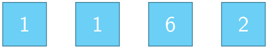
$$\begin{aligned}
&\min\ \beta-\alpha \quad \text{s.t.} \\
&\sum_{j=1}^{n}\sum_{i=1}^{k} x_i z_{ij} - \sum_{i=1}^{k-1} y_i \le \beta && \text{for } 1 \le k \le n \\
&\sum_{j=1}^{n}\sum_{i=1}^{k} x_i z_{ij} - \sum_{i=1}^{k} y_i \ge \alpha && \text{for } 1 \le k \le n \\
&Z\mathbb{1} \le \mathbb{1} \\
&\mathbb{1}^T Z \le \mathbb{1}^T \\
&z_{ij} \in \{0,1\} && \text{for } 1 \le i,j \le n \\
&\alpha,\beta \in \mathbb{R}
\end{aligned}$$

# Principle of the Iterative Rounding algorithm

**Algorithm 1: Iterative Rounding (Rajković 2022)**

For each slot :

- Compute the LP relaxation while fixing the item in the current slot for each remaining item
- Assign the most promising item to this slot according to the LP relaxations




| Day | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| + | | | | |
| - | 5 | 0 | 2 | 3 |

# Principle of the Iterative Rounding algorithm

**Algorithm 1: Iterative Rounding (Rajković 2022)**

For each slot :

- Compute the LP relaxation while fixing the item in the current slot for each remaining item
- Assign the most promising item to this slot according to the LP relaxations

| | | |
|---|---|---|
| 1 | 6 | 2 |

| Day | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| + | 1 | | | |
| - | 5 | 0 | 2 | 3 |

# Principle of the Iterative Rounding algorithm

**Algorithm 1: Iterative Rounding (Rajković 2022)**

For each slot :

- Compute the LP relaxation while fixing the item in the current slot for each remaining item
- Assign the most promising item to this slot according to the LP relaxations

1 1 6 2

7

| Day | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| + | | | | |
| - | 5 | 0 | 2 | 3 |

# Principle of the Iterative Rounding algorithm

**Algorithm 1: Iterative Rounding (Rajković 2022)**

For each slot :

- Compute the LP relaxation while fixing the item in the current slot for each remaining item
- Assign the most promising item to this slot according to the LP relaxations

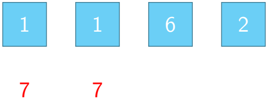

| Day | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| + | 1 | | | |
| - | 5 | 0 | 2 | 3 |

# Principle of the Iterative Rounding algorithm

**Algorithm 1: Iterative Rounding (Rajković 2022)**

For each slot :

- Compute the LP relaxation while fixing the item in the current slot for each remaining item
- Assign the most promising item to this slot according to the LP relaxations




| Day | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| + | | | | |
| - | 5 | 0 | 2 | 3 |

# Principle of the Iterative Rounding algorithm

**Algorithm 1: Iterative Rounding (Rajković 2022)**

For each slot :

- Compute the LP relaxation while fixing the item in the current slot for each remaining item
- Assign the most promising item to this slot according to the LP relaxations

| | | | |
|---|---|---|---|
| 1 | 1 | 6 | 2 |
| 7 | 7 | 6 | 9 |

| Day | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| + | | | | |
| - | 5 | 0 | 2 | 3 |

# Principle of the Iterative Rounding algorithm

**Algorithm 1: Iterative Rounding (Rajković 2022)**

For each slot :

- Compute the LP relaxation while fixing the item in the current slot for each remaining item
- Assign the most promising item to this slot according to the LP relaxations


| Day | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| + | 6 | | | |
| - | 5 | 0 | 2 | 3 |

# Principle of the Iterative Rounding algorithm

**Algorithm 1: Iterative Rounding (Rajković 2022)**

For each slot :

- Compute the LP relaxation while fixing the item in the current slot for each remaining item
- Assign the most promising item to this slot according to the LP relaxations

1 2

| Day | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| + | 6 | 1 | | |
| - | 5 | 0 | 2 | 3 |

# Principle of the Iterative Rounding algorithm

**Algorithm 1: Iterative Rounding (Rajković 2022)**

For each slot :

- Compute the LP relaxation while fixing the item in the current slot for each remaining item
- Assign the most promising item to this slot according to the LP relaxations

1 1 2

6

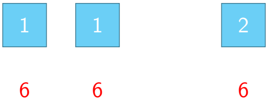
| Day | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| + | 6 | | | |
| - | 5 | 0 | 2 | 3 |

# Principle of the Iterative Rounding algorithm

**Algorithm 1: Iterative Rounding (Rajković 2022)**

For each slot :

- Compute the LP relaxation while fixing the item in the current slot for each remaining item
- Assign the most promising item to this slot according to the LP relaxations




| Day | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| + | 6 | | | |
| - | 5 | 0 | 2 | 3 |

# Principle of the Iterative Rounding algorithm

**Algorithm 1: Iterative Rounding (Rajković 2022)**

For each slot :

- Compute the LP relaxation while fixing the item in the current slot for each remaining item
- Assign the most promising item to this slot according to the LP relaxations


| Day | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| + | 6 | 1 | | |
| - | 5 | 0 | 2 | 3 |

# Principle of the Iterative Rounding algorithm

**Algorithm 1: Iterative Rounding (Rajković 2022)**

For each slot :

- Compute the LP relaxation while fixing the item in the current slot for each remaining item
- Assign the most promising item to this slot according to the LP relaxations

| Day | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| + | 6 | 1 | 1 | 2 |
| - | 5 | 0 | 2 | 3 |

# Results

**Proposition 1: Lower bound**

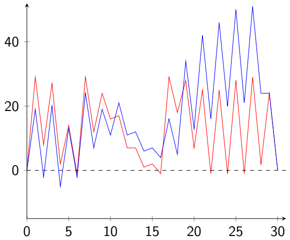
The Iterative Rounding algorithm for the Gasoline Problem has an approximation ratio greater or equal to 2.

NB : there exists already a 2-approximation for the Gasoline problem

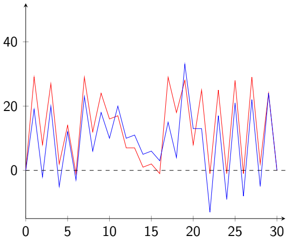
**Proposition 2: Upper bound**

The Greedy algorithm for the $\{1, K\}$-Gasoline Problem has an approximation ratio of 2.

# Experiments : a quick remark



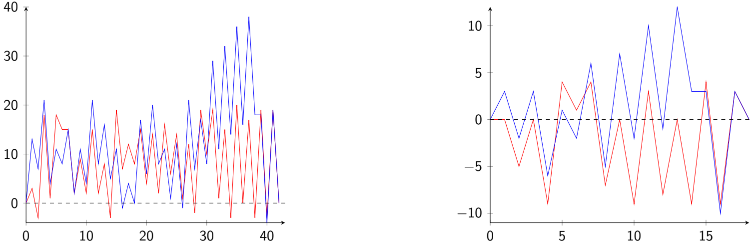

$X = [12, 10, 1, 1, 12, 19, 18, 30, 22, 26, 30, 29, 30, 29, 0]$
$Y = [21, 25, 15, 17, 8, 10, 6, 3, 11, 21, 26, 26, 29, 27, 24]$




$X' = [12, 10, 1, 1, 12, 19, 17, 30, 22, 26, 30, 29, 30, 29, 0]$
$Y' = [21, 25, 15, 17, 8, 10, 6, 3, 11, 20, 26, 26, 29, 27, 24]$

Left : 1.86, Right : 1.57

# Experiments : Local Search



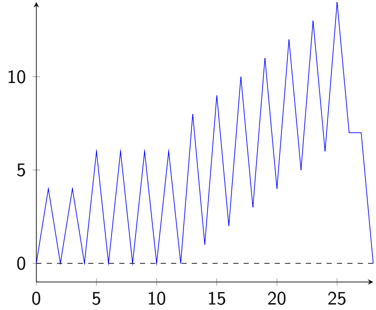

Instances found by Local Search with common characteristics

# Instance construction for the lower bound

Staircase instance ($a_i \leq m-1$) :

$$\begin{aligned} X &= [a_0\,, \cdots, a_k\,, \quad m \quad, \cdots, \quad m \quad, \quad 0 \quad] \\ Y &= [a_0\,, \cdots, a_k\,, m-1\,, \cdots, m-1\,, m-1] \end{aligned}$$

IR always yields the identity permutation : value $= 2(m-1)$
$\hookrightarrow$ find the $(a_i)_i$ so that the optimal value is around m.

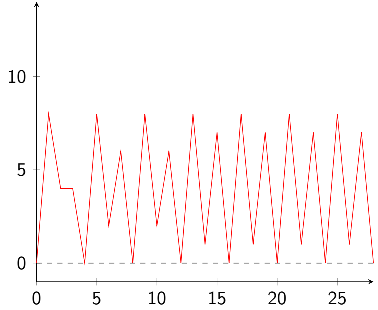


# Instance construction for the lower bound

Staircase instance ($a_i \leq m-1$) :

$$
\begin{aligned}
X &= [a_0\,, \cdots, a_k\,, \quad m \quad, \cdots, \quad m \quad, \quad 0 \quad ] \\
Y &= [a_0\,, \cdots, a_k\,, m-1\,, \cdots, m-1\,, m-1]
\end{aligned}
$$

IR always yields the identity permutation : value $= 2(m-1)$
$\hookrightarrow$ find the $(a_i)_i$ so that the optimal value is around m.



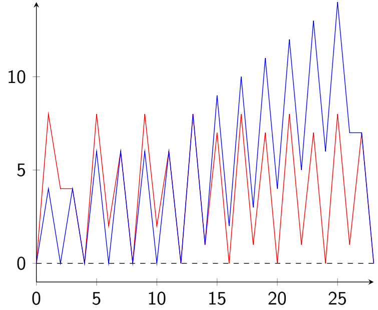
# Instance construction for the lower bound

Staircase instance ($a_i \leq m-1$) :

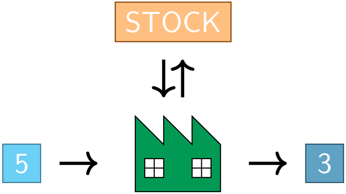
$$X = [a_0, \cdots, a_k, m, \cdots, m, 0]$$
$$Y = [a_0, \cdots, a_k, m-1, \cdots, m-1, m-1]$$

IR always yields the identity permutation : value $= 2(m-1)$

$\hookrightarrow$ find the $(a_i)_i$ so that the optimal value is around m.

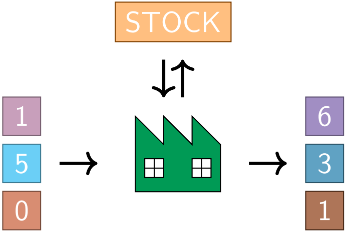


# Back to the example...




| Day | + | - |
|---|---|---|
| 1 | 6 | 5 |
| 2 | 1 | 0 |
| 3 | 1 | 2 |
| 4 | 2 | 3 |

# Back to the example...

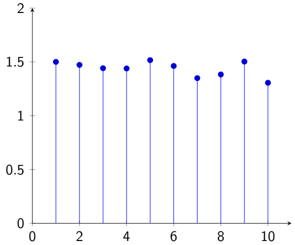


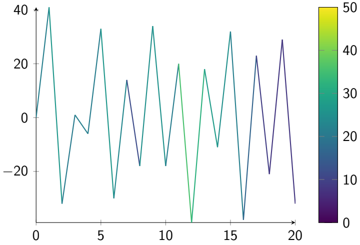

| Day | + | - |
| --- | --- | --- |
| 1 | 1 | 6 |
| | 5 | 3 |
| | 0 | 1 |
| 2 | 2 | 4 |
| | 2 | 1 |
| | 2 | 7 |

# Experiments

Local Search adapted to the generalised problem

- In previous works, conjecture on an approximation factor of 2
- Experiments for $d \leq 10$ tends to confirm this



# Result

**Proposition 3**

The Iterative Rounding algorithm for the Generalised Gasoline problem has an approximation ratio greater or equal than 2.

Comes directly from the lower bound on 1D version (0 everywhere except on the 1st coordinate)

# Recap and further work

- No better approximation with IR algorithm
- Still interesting to consider for higher dimensions
- Higher dimensions seem not to be harder for the IR algorithm

Further possible works:

- Prove guaranties for the IR algorithm for $\{1, K\}$ and general cases
- Identify other subcases where a greedy algorithm has a constant approximation ratio
- Understand whether augmenting the dimension creates worse instances (lower bound)

# Iterative Rounding algorithms for the generalised Gasoline Problem

Lucas Lorieau

Grenoble INP, Ensimag - UGA

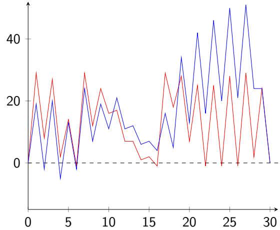
June 26, 2024

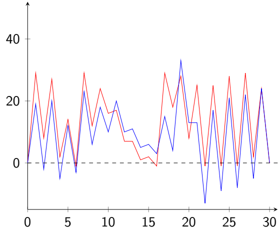
Instance for which Greedy approximates arbitrarily badly :

$$X = \{\underbrace{2,\dots,2}_{\frac{n}{4}\text{ times}},\underbrace{1,\dots,1}_{\frac{n}{2}\text{ times}},\underbrace{0,\dots,0}_{\frac{n}{4}\text{ times}}\},$$

$$Y = \{\underbrace{2,0,\dots,2,0}_{\frac{n}{2}\text{ times}},\underbrace{2,\dots,2}_{\frac{n}{4}\text{ times}},\underbrace{0,\dots,0}_{\frac{n}{4}\text{ times}}\}.$$

Left : 1.86, Right : 1.57




$$X = [12, 10, 1, 1, 12, 19, \textcolor{blue}{18}, 30, 22, 26, 30, 29, 30, 29, 0]$$
$$Y = [21, 25, 15, 17, 8, 10, 6, 3, 11, \textcolor{blue}{21}, 26, 26, 29, 27, 24]$$

$$X' = [12, 10, 1, 1, 12, 19, \textcolor{red}{17}, 30, 22, 26, 30, 29, 30, 29, 0]$$
$$Y' = [21, 25, 15, 17, 8, 10, 6, 3, 11, \textcolor{red}{20}, 26, 26, 29, 27, 24]$$

$$\min \|\beta - \alpha\| \qquad \text{s.t.}$$

$$\sum_{j=1}^{n} \sum_{i=1}^{k} x_i^{(p)} z_{ij} - \sum_{i=1}^{k-1} y_i^{(p)} \leq \beta^{(p)} \qquad \text{for } 1 \leq k \leq n, \forall p \leq l$$

$$\sum_{j=1}^{n} \sum_{i=1}^{k} x_i^{(p)} z_{ij} - \sum_{i=1}^{k} y_i^{(p)} \geq \alpha^{(p)} \qquad \text{for } 1 \leq k \leq n, \forall p \leq l$$

$$Z\mathbb{1} \leq \mathbb{1}$$

$$\mathbb{1}^T Z \leq \mathbb{1}^T$$

$$z_{ij} \in \{0, 1\} \qquad \text{for } 1 \leq i, j \leq n$$

$$\alpha, \beta \in \mathbb{R}^l$$

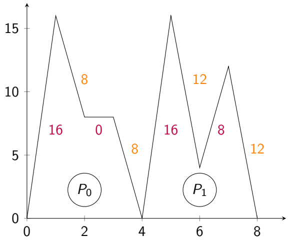
| n | Max | Mean | $\sigma$ | % of non-optimal |
|---|---|---|---|---|
| 5 | 1.5 | 1.036 | 0.006 | 6.77 |
| 10 | 1.667 | 1.085 | 0.011 | 18.30 |
| 15 | 1.667 | 1.100 | 0.012 | 22.06 |
| 20 | 1.667 | 1.111 | 0.013 | 24.89 |

IR yields the identity permutation as solution:

$$\begin{pmatrix} \color{red}{I_{k+1}} & 0 \\ 0 & \color{blue}{\frac{1}{m}J_m} \end{pmatrix} \quad \text{where } J_m = \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{pmatrix}$$

Value : $2(m-1)$

Goal : find $(a_i)$ so that the optimal value is around $m$




Pattern $P_i = (m, u_{i+1}, u_i, u_{i+1})$ with $m + u_i = 2u_{i+1}$ and $m = 2^p$